

A Prototype File Protocol for Application Data Sets Based on HDF

Kent E. Eschenberg, Ph.D.

U.S. Army Engineer Research and Development
Center Major Shared Resource Center, Vicksburg, MS
kee@erdc.hpc.mil

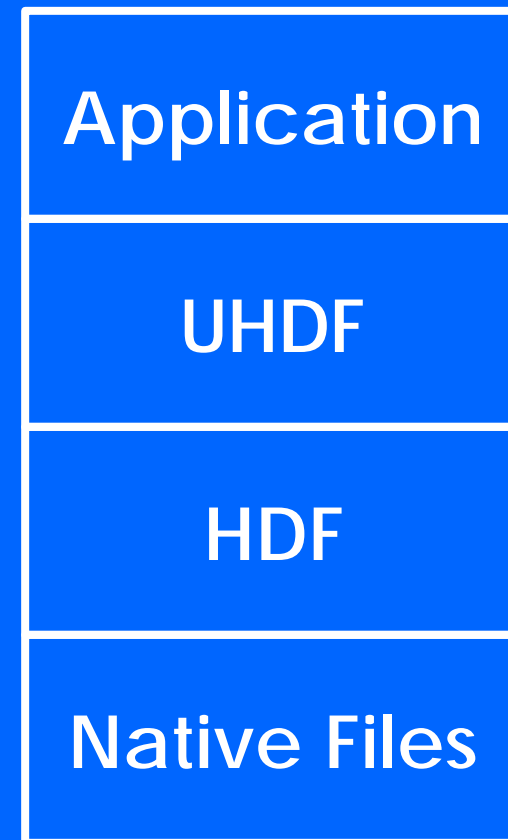
Mike Folk, Ph.D.

National Center for Supercomputer Applications,
University of Illinois at Urbana-Campaign
mfolk@ncsa.uiuc.edu

Summary

The Unstructured HDF (UHDF) file protocol is a layer of software between an application and the HDF5 file protocol that adds:

- Support for time-varying data
- Support for unstructured data sets
- Metadata that uniquely identifies each object
- Metadata about the data



Why HDF?

The “Hierarchical Data Format”, Version 5, is based on earlier versions developed at NCSA over many years.

- Its generic with support for grouping and many data types
- Flexible access and storage methods
- Support for many platforms with binary conversion
- Designed for large HPC data sets and I/O needs
- Used on many large projects including ASCI

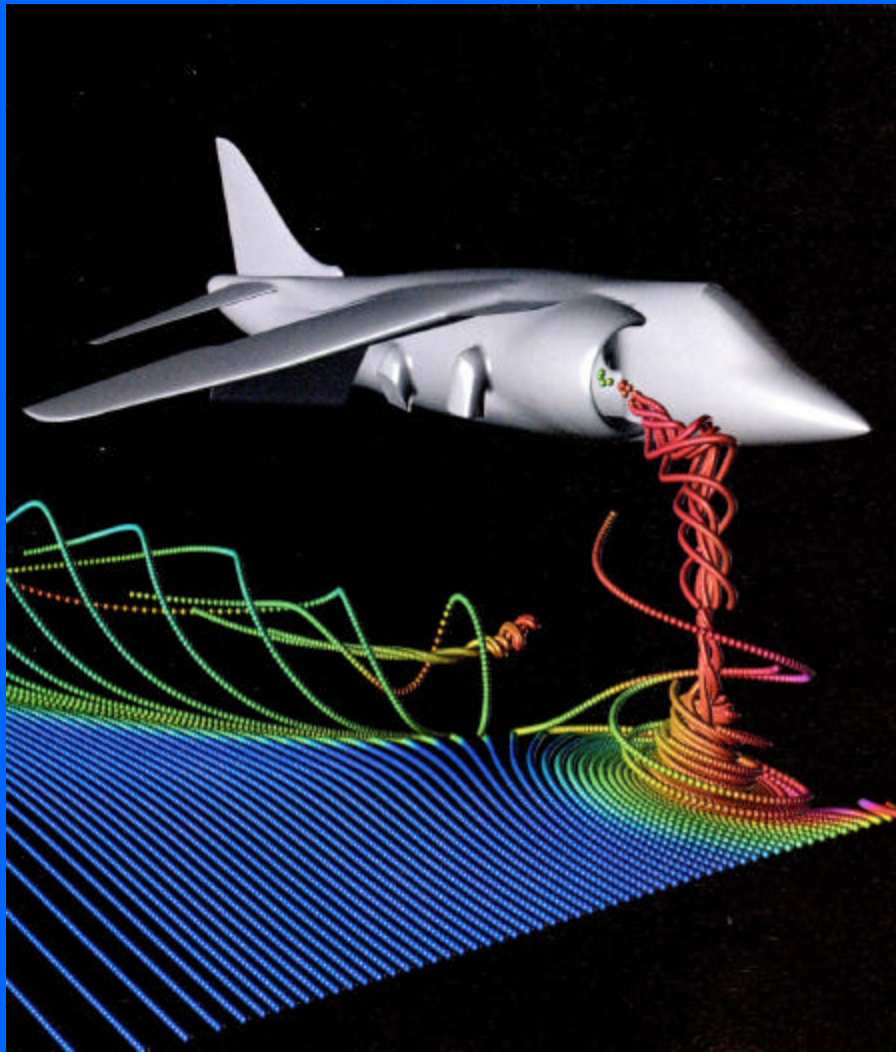
Organization

- An hierarchy of *objects*
- Each object can be a data array or a group of objects
- Modest amounts of data can be attached to any object as *attributes*
- A data array can be of many types (int, float, double, strings, etc.)

Remaining Needs

- Storage for time-varying and unstructured data
- Unique identification of each object
- A standard storage and API for these additional features

Application Data Sets



NASA
Computer Graphics World
June 2001

Basic Types

- Structured
- Unstructured

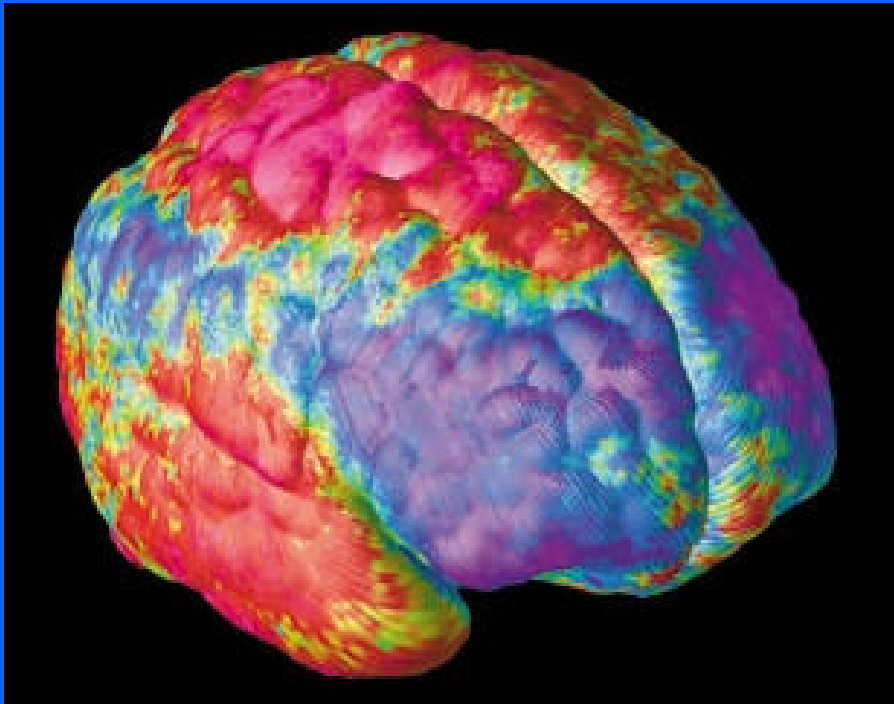
Structured

- 2D or 3D arrays
- Number of nodes along X is the same everywhere
- Likewise for Y and (if 3D) for Z

Example

- Structured 3D array
- Node data records velocity

Application Data Sets (cont.)



UCLA Laboratory
of Neuro Imaging
National Institute of Mental Health
Computer Graphics World
February 2001

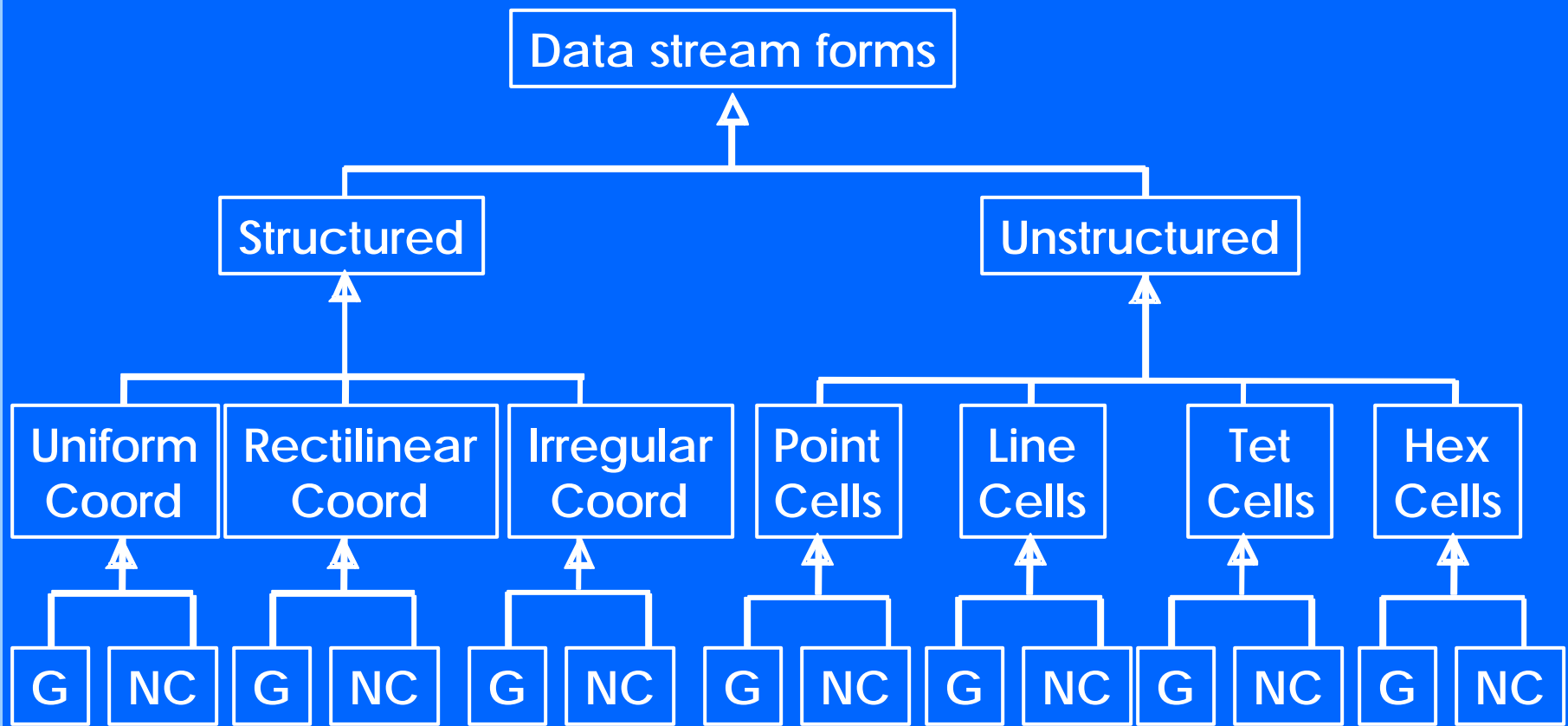
Unstructured

- 1D array of cells
- Cell types include
 - o point, line
 - o triangle, quad
 - o tetrahedron
 - o hexahedron

Example

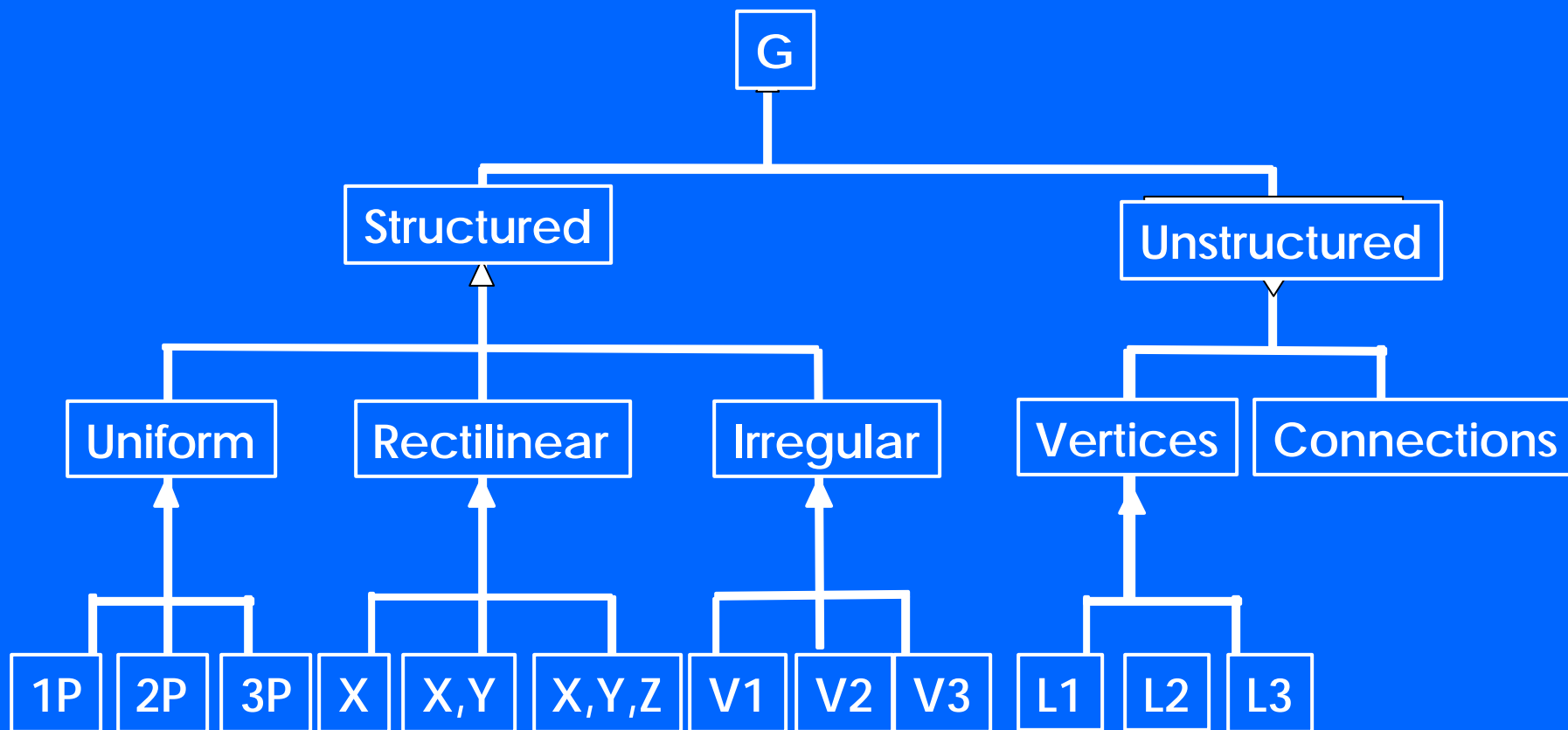
- Unstructured cells
- Cell data records tissue loss due to schizophrenia

Hierarchy of Data Set Types



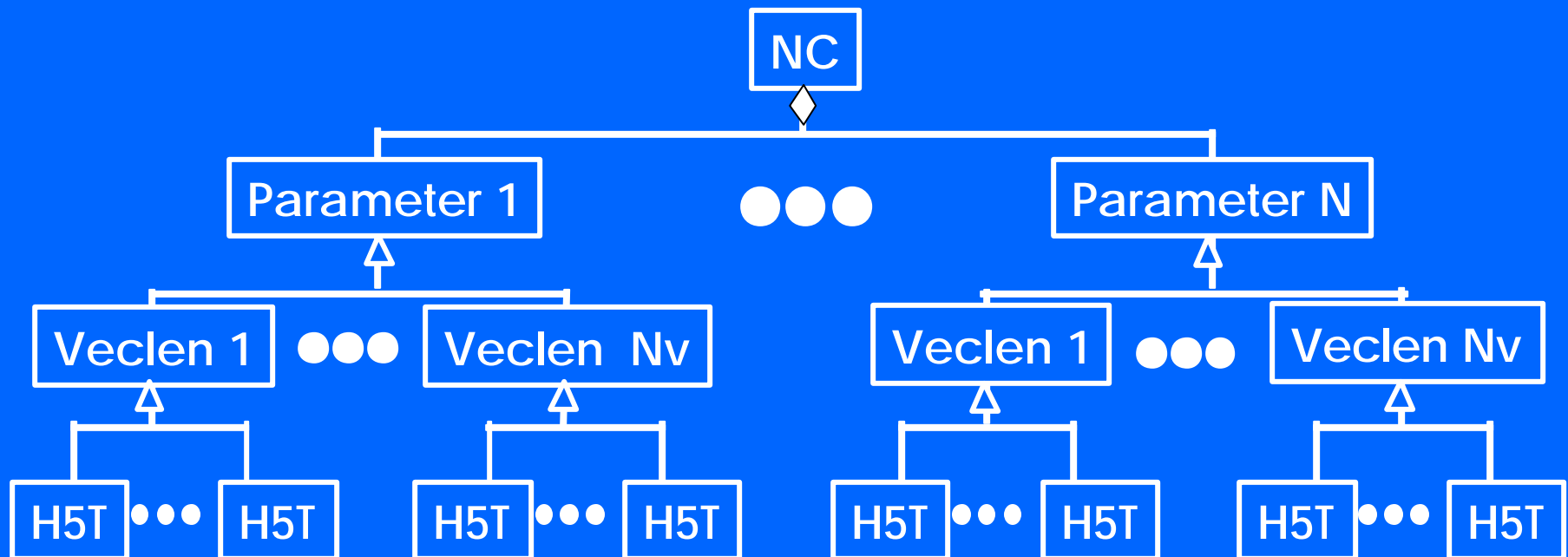
Hierarchy of Data Set Types (cont.)

- Grid (mesh) data types



Hierarchy of Data Set Types (cont.)

- Node and cell data types



UHDF organizes the information into 4 sections:

(1) General Metadata (an HDF5 group object)

- About data source
- About previous postprocessing
- About this file
- Notes

(2) Datamap Metadata (an HDF5 group object)

- Collects objects into *streams*
- Identifies role of object (vertices, node data, etc.)
- Stores hints (ex: min/max values)
- Includes list of times for which object is stored

Basic UHDF Approach (cont.)

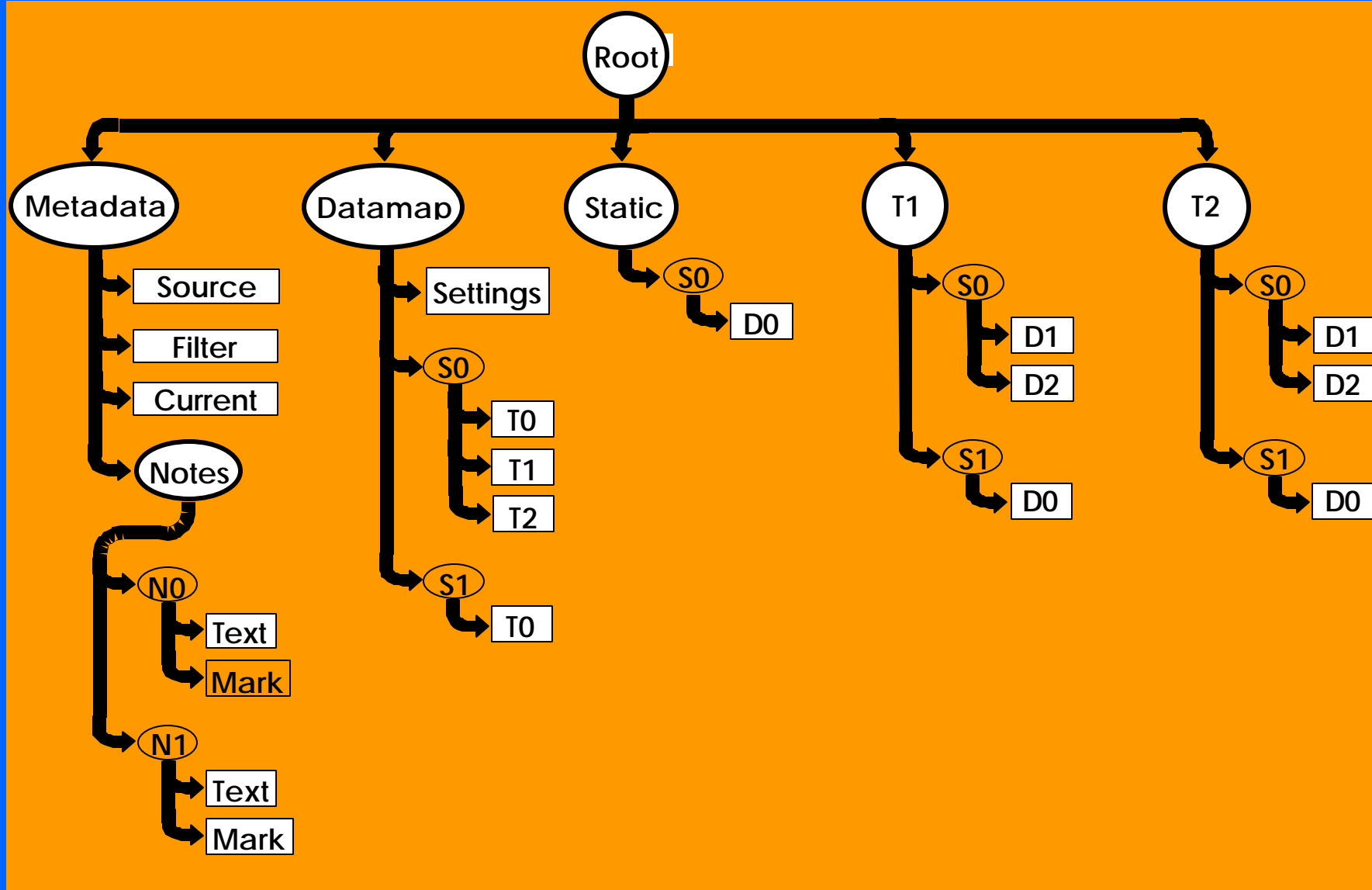
(3) Static Data Object (an HDF5 group object)

- Any object that is time invariant

(4) Dynamic Data (a series of HDF5 group objs)

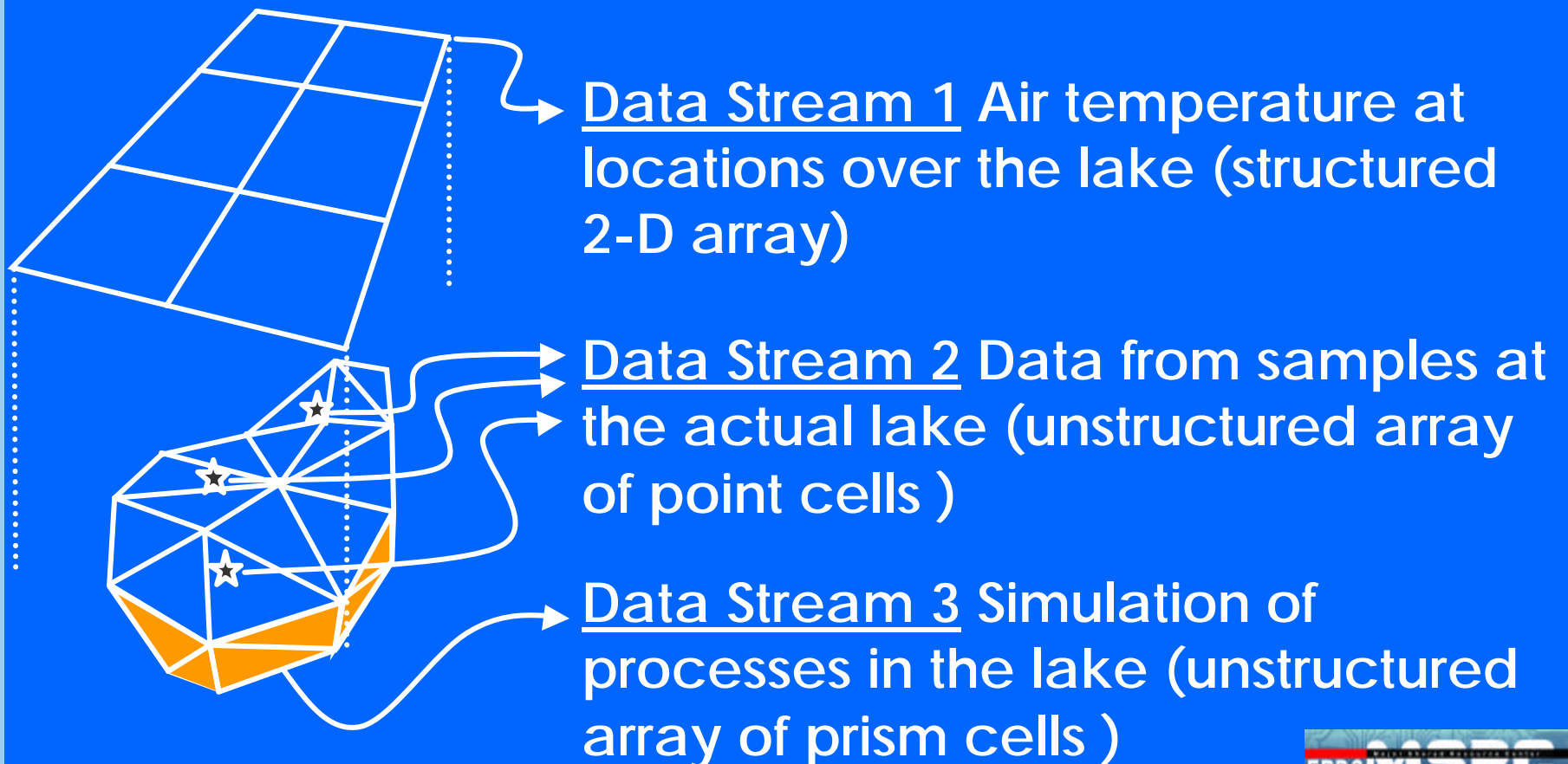
- One per time step
- Each contains all objects defined for that time
- Objects can be defined for some times and not others

Basic UHDF Approach (cont.)



Data Streams

Data streams, in effect, combine multiple files into one. Each stream has its own form, vertices, data, time steps.



Proposed Project Goals

- Refine design with input from potential users
- Map information structures of common packages (VTK, EnSight, etc.) to UHDF structures
- Create UHDF library
- Create APIs for C, C++, Fortran
- Create readers for AVS, EnSight, VTK, OpenDX
- Create utilities that can convert between UHDF files and those of AVS, EnSight, VTK, OpenDX
- Create documentation, tutorials

How to Get Involved

- Contact the authors of this paper